

# Software-Projekt I

Prof. Dr. Rainer Koschke

Arbeitsgruppe Softwaretechnik  
Fachbereich Mathematik und Informatik  
Universität Bremen

Sommersemester 2013

## Planung eines Software-Projekts I

### Planung

Projekt

Vorgehen

Inhalt

Zeitplan

Planung und Aufwand im SWP 05/06

Risiken

Risiko-Management

Erfahrungen aus dem SWP 05/06

Allgemeine Risiken in einem Software-Projekt

Wiederholungsfragen



- Was ist ein Projekt?
- Wann wird in einem Projekt geplant?

## Projekt und Plan

### Definition

**Projekt:** eine für begrenzte Zeit mit bestimmtem Ziel bestehende Organisation mit all ihren Bestandteilen, Beziehungen im Innern und nach außen.

### Definition

**Plan:** gedankliche Vorwegnahme zukünftigen Handelns.

### Definition

😊 **Projektplan:** Versuch, Zufall gegen Irrtum zu ersetzen (in der Praxis wird oft Irrtum zum Zufall hinzuaddiert).

## Planung

- am Anfang:
  - Gliederung in Phasen, Aktivitäten, Arbeitspakete,
  - zeitlicher Ablauf (Meilensteine),
  - Arbeitsorganisation,
  - Aufbau der Dokumentation.
- während des Projekts:
  - Controlling (d.h. die Überwachung des Projektfortschritts),
  - Reaktion auf Abweichung (Anpassung des Plans).

Planung hat im Software-Projekt zwei verschiedene Bedeutungen: Zu Beginn des Projekts müssen Pläne aufgestellt werden für die Gliederung in Phasen, den für die Arbeitsorganisation, den für den Aufbau der Dokumentation, und den für die Prüfungen. Die Existenz und die Qualität dieser Pläne hat zentrale Bedeutung für den Erfolg des Projekts.

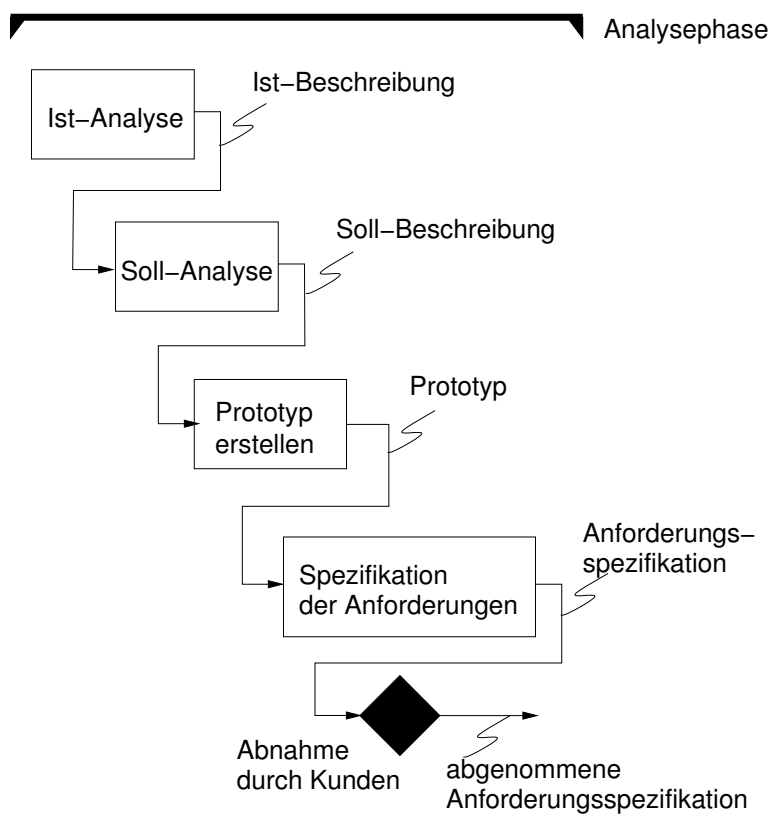
Während des Projekts bleibt die Planung eine Daueraufgabe, weil sich die Realität meist nicht an unsere Planungen hält. Notwendig ist ein Kompromiss zwischen der starrsinnigen Verteidigung von Plänen, die offenkundig nicht einzuhalten sind, und der laufenden Anpassung der Pläne an den realen Stand. Eine gewisse Spannung zwischen Plan und Realität ist normal. Planung und Planverfolgung sind die wichtigsten Aufgaben der Projektleitung. Eng damit verbunden sind Controlling (d.h. die Überwachung der Ausgaben in einem Projekt) und Qualitätssicherung (d.h. die Prüfung aller Teilresultate und Resultate).



Was sind die Konzepte eines Projektplans?

6 / 58

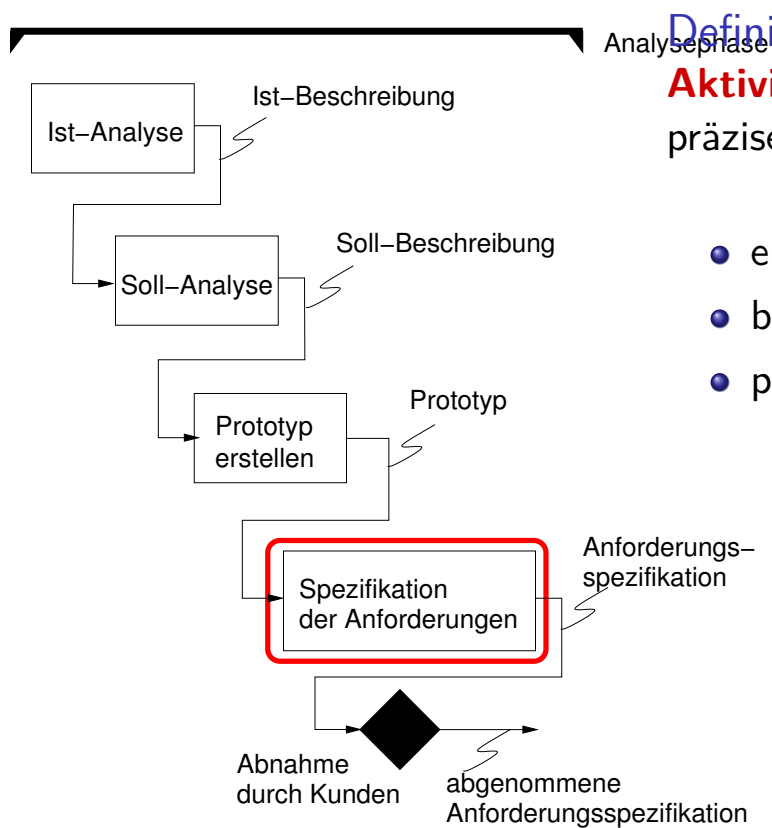
## Elemente eines Projekts



7 / 58

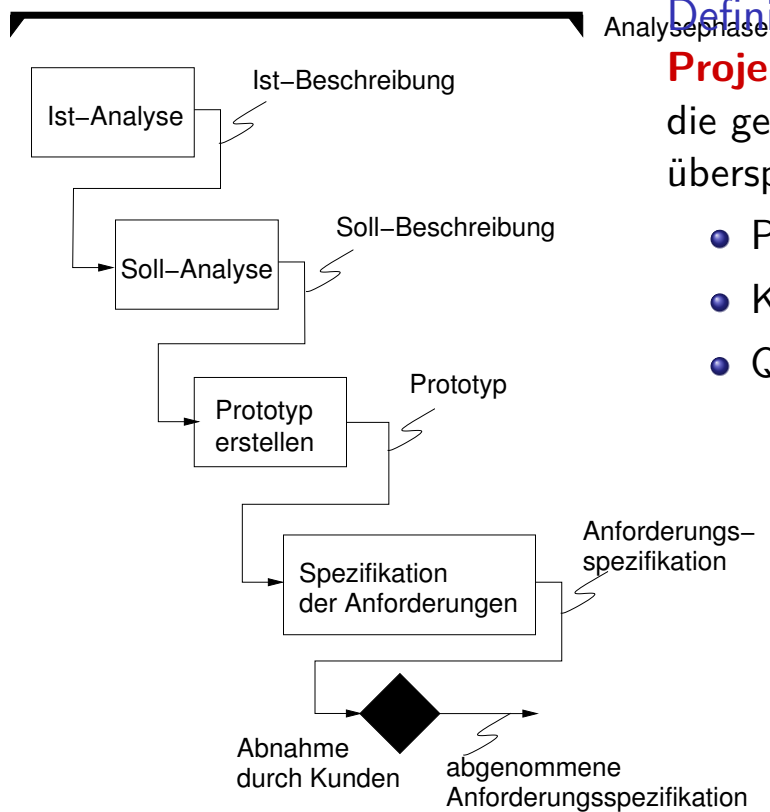
Für das Verständnis von Projekten sind einige Grundbegriffe notwendig. Keiner dieser Begriffe ist spezifisch für ein Software-Projekt. Insgesamt kann angemerkt werden, dass allgemeine Aussagen zu Projekten (jedweder Form) in aller Regel auch für Software-Projekte gelten. Die Natur der Software, die in einem Software-Projekt entwickelt werden soll, kommt vielmehr in den Inhalten der Arbeitspakete zum Tragen.

## Elemente eines Projekts



- enthält Aufgaben
- benötigt Ressourcen
- produziert Arbeitsprodukt

# Elemente eines Projekts



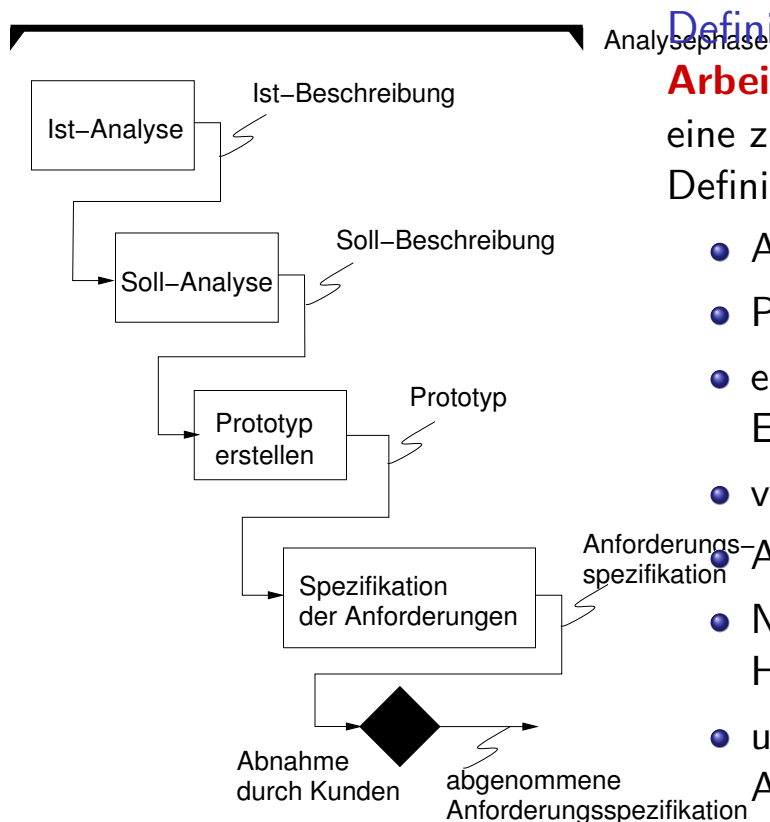
## Definition

**Projektfunktion:** Aktivität, die die gesamte Projektlaufzeit überspannt. Beispiele:

- Projektmanagement
- Konfigurationsmanagement
- Qualitätssicherung.

9 / 58

# Elemente eines Projekts



## Definition

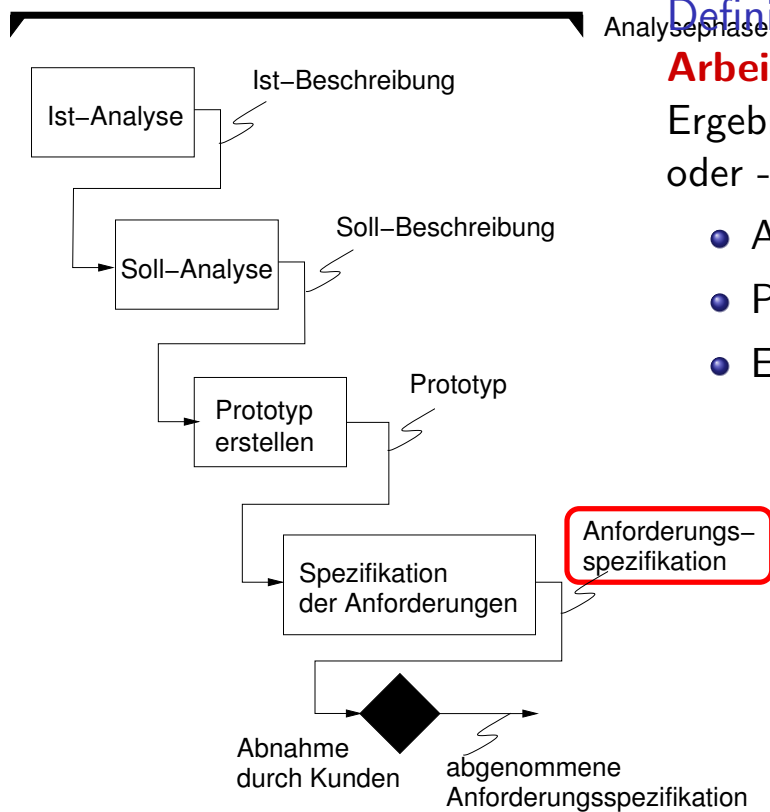
**Arbeitspaket:** Spezifikation für eine zu erledigende Arbeit.

Definiert

- Arbeitsprodukt,
- Personalbedürfnisse,
- erwartete Entwicklungsdauer,
- verwendete Ressourcen,
- Akzeptanzkriterien,
- Namen des Hauptverantwortlichen
- und sonstige relevante Aspekte der Arbeit.

10 / 58

# Elemente eines Projekts



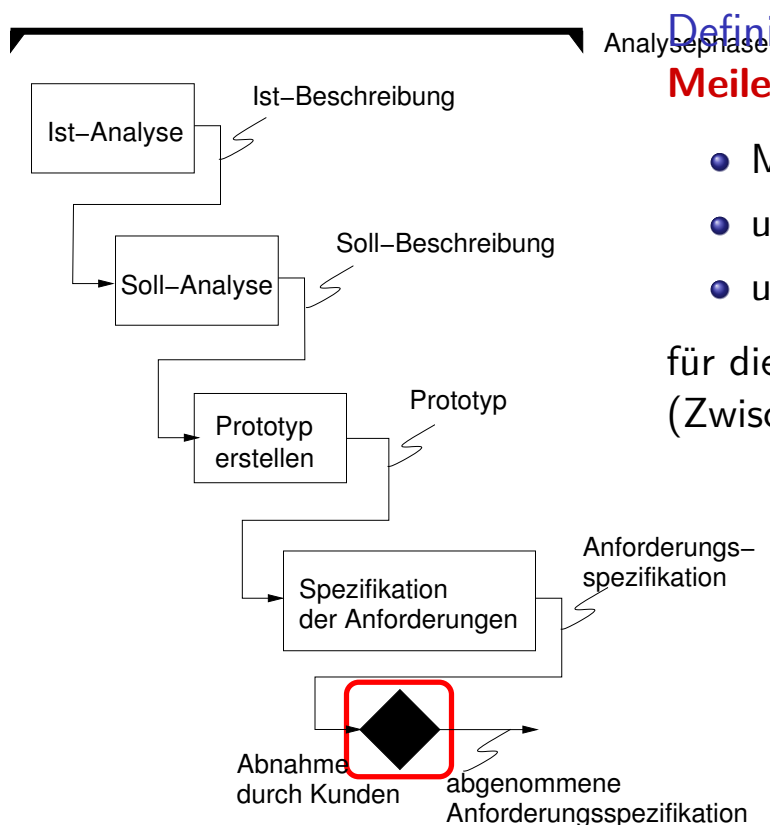
## Definition

**Arbeitsprodukt** ein konkretes Ergebnis einer Projektfunktion oder -aktivität. Beispiele:

- Anforderungsspezifikation,
- Projektplan,
- Entwurfsdokumente

11 / 58

# Elemente eines Projekts



## Definition

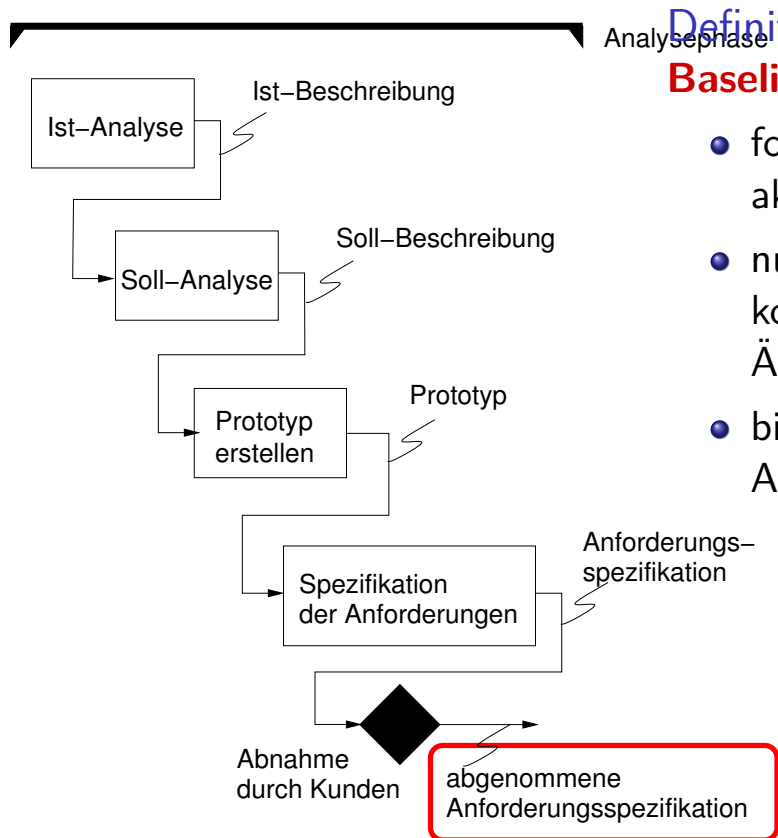
### Meilenstein:

- Menge von Kriterien
- und vorgesehener Zeitpunkt
- und erreichter Zeitpunkt

für die Abnahme eines (Zwischen-)Resultats

12 / 58

# Elemente eines Projekts



## Definition

**Baseline:** Arbeitsprodukt,

- formal begutachtet und akzeptiert,
- nur durch formalen kontrollierten Änderungsprozess änderbar;
- bildet Basis für nachfolgende Arbeitsaktivitäten.

13 / 58

## Fragen



Welche Pläne erstelle ich?  
Wie gehe ich bei der Planung vor?

14 / 58



# Vorgehen zur Planung

Iteration dieser Schritte, bis Resultate zusammenpassen:

- ① Auswahl eines geeigneten Prozesses (oder Prozessmodells)
- ② Abschätzung des Aufwands sowie der Risiken
- ③ Zuordnung der Ressourcen und Aufstellen eines Zeitplans
- ④ Aufstellen eines Dokumentationsplans
- ⑤ Aufstellen eines Prüfplans
- ⑥ Aufstellen eines Organisationsplans
- ⑦ Definition der Meilensteine (1-6)

15 / 58

## Vorgehen zur Planung

Wo es definierte Prozessmodelle gibt, kann (und muss) die Projektleiterin die Vorlagen für die Pläne aus der Schublade (d.h. aus dem Intranet) nehmen. Das Prozessmodell legt die Aktivitäten, ihre Arbeitsprodukte sowie deren Abhängigkeiten fest.

Dokumentationsplan: (**inhaltlich**) Anforderungen an die Dokumentation, Meilensteine, Baselines, Reviews und Abnahme der Software-Dokumentation; (**formal**) Style-Guide, Namenskonventionen, Dokumentationsformate

Prüfplan: Wie werden Arbeitsergebnisse und Projektfortschritt überprüft?

Der Projektplan sollte sorgfältig geprüft werden!

Der Projektplan entsteht mit dem Projekt und wird im Verlauf des Projekts fortgeschrieben, aber nicht laufend geändert.

Der Projektplan ist Gegenstand der Änderungskontrolle.



Was ist der Inhalt eines Projektplans?

16 / 58

## Inhalt des Projektplans

Plan macht prinzipiell Aussagen zu den folgenden W-Punkten:

- warum
- was getan wird,
- für wieviel Geld,
- von wem,
- wann und
- womit, d.h. unter Einsatz welcher Hilfsmittel und Techniken.

17 / 58

# Aufbau eines Projektplans nach IEEE-Std-1058 (1987)

## Übersicht

0. Version und Änderungsgeschichte
1. Einleitung
2. Projektorganisation
3. Managementprozess
4. Technischer Prozess
5. Arbeitspakete, Zeitplan und Budget
6. Zusätzliche Elemente
7. Index
8. Anhang

18 / 58

Der Projektplan, wie jedes andere Dokument auch, erhält eine Versionsnummer. Änderungen des Projektplans sind unausweichlich (er wird schrittweise verfeinert und angepasst). Die Änderungen müssen dokumentiert und nachvollziehbar sein. Der Abschnitt "Version und Änderungsgeschichte" enthält diese Information.

Die Einleitung gibt einen Überblick über das Projekt und das Produkt, die Liste der auszuliefernden Dinge, den Plan für die Entwicklung und die Fortführung des Projektplans, Referenzmaterial sowie Definitionen und Akronyme.

Der Abschnitt "Projektorganisation" spezifiziert das Prozessmodell des Projekts, die Organisationsstruktur, die Grenzen und Schnittstellen der Organisation und die individuellen Verantwortlichkeiten.

Der Abschnitt "Managementprozess" spezifiziert die Managementziele und -prioritäten, Annahmen und Einschränkungen, Risiken und deren Behandlung sowie Kontrollmechanismen und den Personalplan.

Der Abschnitt "Technischer Prozess" spezifiziert die technischen Methoden, Werkzeuge und Techniken, die im Projekt benutzt werden sollen. Außerdem werden der Dokumentationsplan und die Pläne für die Projektfunktionen spezifiziert, wie etwa die Qualitätssicherung und das Konfigurationsmanagement.

Der Abschnitt "Arbeitspakete" spezifiziert die Arbeitspakete, ihre Abhängigkeiten und Beziehungen, die benötigten Ressourcen, Zuteilung des Budgets und Ressourcen auf Arbeitspakete sowie den Zeitplan.

Der Abschnitt "Zusätzliche Elemente" enthält Pläne für zusätzliche Komponenten.

## Aufbau eines Projektplans nach IEEE-Std-1058 (1987)

### Einleitung

#### 1.1 Projektübersicht

Zusammenfassung der Ziele, Resultate, Hauptarbeitsaktivitäten und -produkte, Hauptmeilensteine, benötigte Ressourcen, grober Zeitplan und Budget; Kontaktdaten des Kunden

#### 1.2 Auszuliefernde Produkte

alle an den Kunden auszuliefernde Produkte mit Auslieferungsdatum und -ort sowie deren Anzahl (die Anforderungsspezifikation ist ein separates Dokument)

#### 1.3 Evolution des Plans

Plan für vorausgesehene und nicht vorausgesehene Aktualisierung des Projektplans und deren Bekanntmachung

#### 1.4 Referenzen

#### 1.5 Definitionen und Akronyme

Die Referenzen enthalten Verweise auf alle Dokumente und andere Informationsquellen, die im Projektplan referenziert werden. Hierfür gelten die üblichen Zitierregeln.

Der Abschnitt "Definitionen und Akronyme" beschreibt alle Begriffe, die für das Verständnis des Projektplans notwendig sind, auch in Form von Verweisen auf entsprechende Dokumente mit deren Definition.

Es handelt sich hier nicht um das Glossar der Anforderungsspezifikation, von dem später noch die Rede sein wird. Der Projektplan kann aber auf dieses Glossar verweisen.

## Projektorganisation

### 2.1 Prozessmodell

Beziehungen zwischen den Projektfunktionen und -aktivitäten mit Hauptmeilensteinen, Baselines, Reviews, Produkte (interne und auszuliefernde) und Abschlüsse

### 2.2 Organisationsstruktur

interne Managementstruktur (z.B. durch Organigramme): Weisungsbefugnis, Verantwortlichkeit und Kommunikation im Projekt (Kontaktdaten aller Beteiligten nicht vergessen!)

### 2.3 Organisationsgrenzen und -schnittstellen

zwischen übergeordneter Organisation, Kundenorganisation und Untervertragsnehmer

### 2.4 Verantwortlichkeiten

Auflistung aller Projektfunktionen und -aktivitäten unter Nennung der Verantwortlichen

3.1 Managementziele und -prioritäten

Beispiele: Häufigkeit und Mechanismen der Berichterstattung; relative Prioritäten zwischen Anforderungen, Zeitplan und Budget; Absicht zur Wiederverwendung existierender Software

3.2 Annahmen, Abhängigkeiten und Einschränkungen

Annahmen, auf denen das Projekt beruht; externe Ereignisse, von denen es abhängt; Beschränkungen, unter denen das Projekt durchgeführt wird

3.3 Risikomanagement

Identifikation und Bewertung von Risiken; Mechanismen für Verfolgung der Risikofaktoren; Notfallpläne;  
Beispiele: Risiken mit Verträgen, technologische Risiken, Größe und Komplexität der Aufgabe, Personal, Akzeptanz des Kunden etc.

Managementprozess (Fortsetzung)

3.4 Projektüberwachung

Berichtswesen, Formate für Berichte, Informationsflüsse, Reviews und Audits; auf der Ebene von Arbeitspaketen; Beziehung zu Projektfunktionen (bspw. Qualitätssicherung, Konfigurationsmanagement)

3.5 Mitarbeiter

Anzahl und Typen der notwendigen Mitarbeiter; erforderliche Fähigkeiten, Beginn und Dauer der Mitarbeit; Methoden zur Anwerbung, Ausbildung, Bindung und Ausgliederung von Mitarbeitern

#### 4.1 Methoden, Werkzeuge und Techniken

Entwicklungsplattform, Entwicklungsmethode, Programmiersprache sowie andere Notationen, Techniken und Methoden, um das System und andere auszuliefernde Produkte zu spezifizieren, entwerfen, konstruieren, testen, integrieren, dokumentieren, modifizieren oder pflegen; technische Standards, Richtlinien, Zertifizierungskriterien

#### 4.2 Dokumentationsplan

Anforderungen an die Dokumentation, Meilensteine, Baselines, Reviews und Abnahme der Software-Dokumentation; Style-Guide, Namenskonventionen, Dokumentationsformate

#### 4.3 Unterstützende Projektfunktionen

z.B. Konfigurationsmanagement und Qualitätssicherung mit Verantwortlichkeiten, Ressourcen, Zeitplänen und Budget

### Arbeitspakete, Zeitplan und Budget

#### 5.1 Arbeitspakete

eindeutig identifizierbar (z.B. mit Nummer);  
Zerlegung der Arbeitspakete

#### 5.2 Abhängigkeiten

Abhängigkeiten zwischen Arbeitspaketen und zu externen Elementen; Reihenfolge der Abarbeitung

#### 5.3 Ressourcenanforderung

Dauer und Ressourcen;  
Beispiele: Anzahl und Qualifikation des Personals, Hardware, unterstützende Software, Büro- und Laborräume, Reisekosten, Wartungskosten

#### 5.4 Zuteilung des Budgets und der Ressourcen auf Projektfunktionen und Aktivitäten

#### 5.5 Zeitplan

Beispiele für zusätzliche Elemente:

- Managementpläne für Unterauftragsnehmer
- Ausbildungspläne
- Beschaffungspläne für Hardware
- Raumpläne
- Installationspläne
- Pläne für die Konvertierung von Daten
- Pläne für die Übergabe des Systems (intern, extern)
- Pläne für die Wartung und Evolution

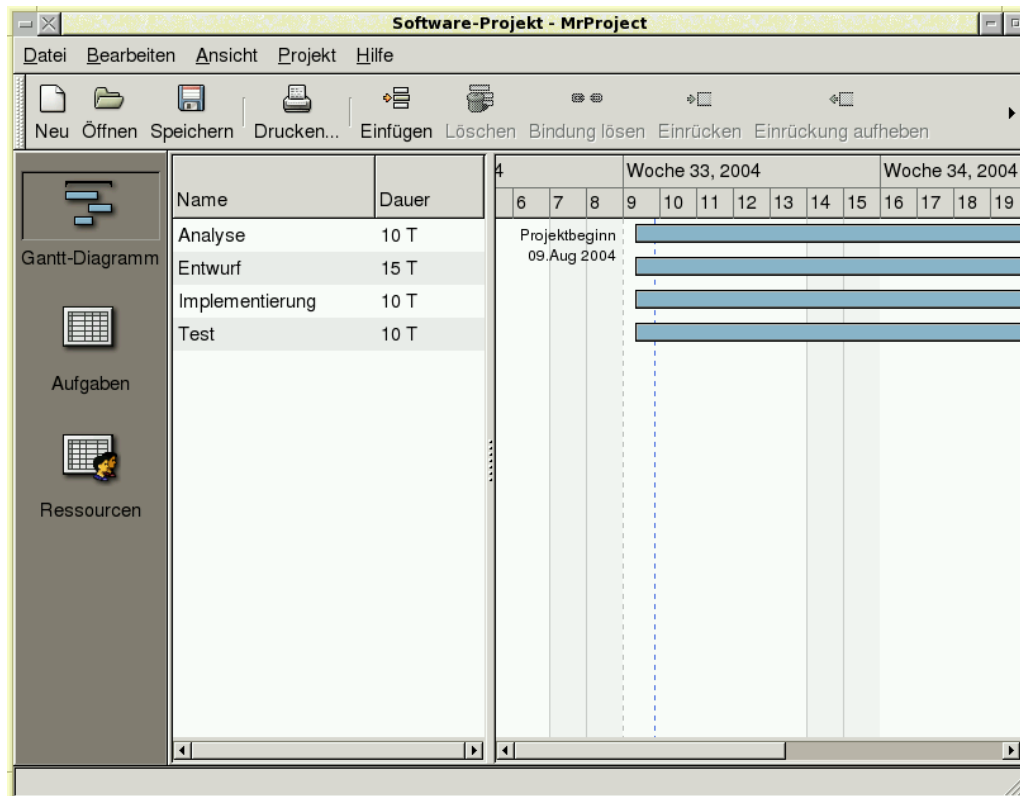
## Fragen



Wie schreibe ich einen Zeitplan auf?



# Zeitplan: Aktivitäten und Dauer (Gantt-Diagramme)



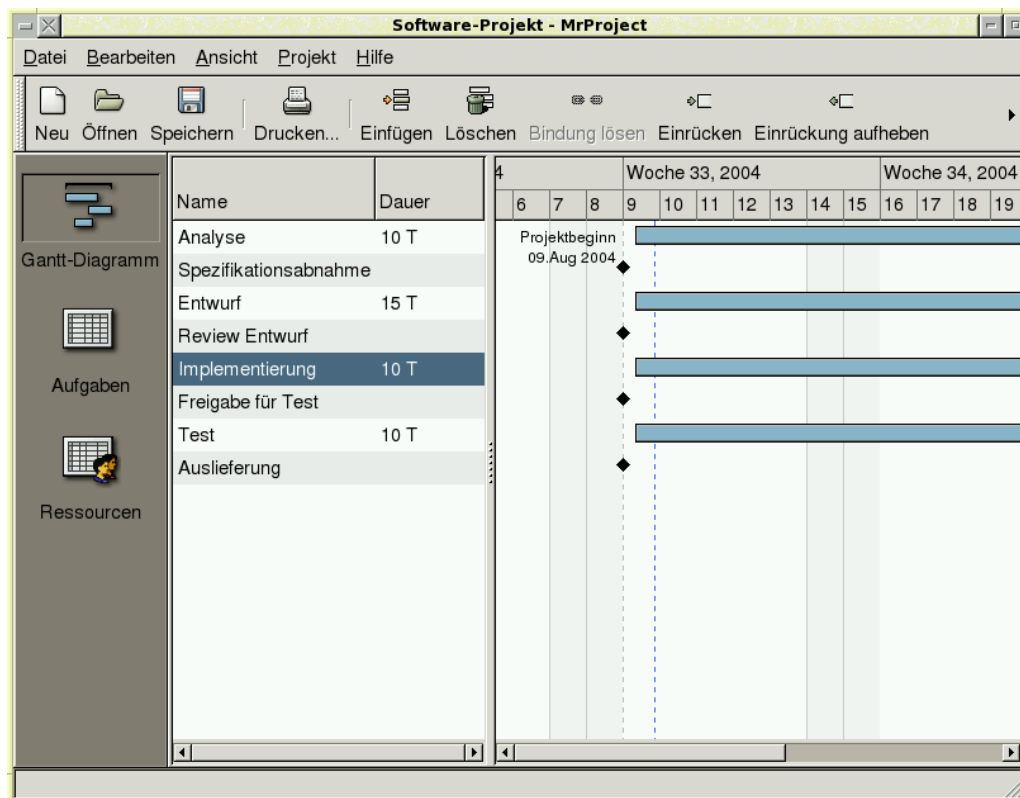
21 / 58

Die Gantt-Notation ist nach ihrem Erfinder, Henry L. Gantt (1917), benannt. Sie beschreibt kausale Abhängigkeiten einzelner Aktivitäten sowie deren zeitlichen Verlauf.

Das Projekt wird zunächst in einzelne Aktivitäten gegliedert. Deren Aufwand wird geschätzt. Mögliche Aktivitäten behandeln wir in späteren Kapiteln.

Die Granularität der Aktivitäten muss angemessen sein. Die in diesem Beispiel gewählte ist es sicher nicht. Je detaillierter die Angabe desto leichter fällt die Schätzung (und desto aufwändiger die Planung; was sich jedoch in der Regel auszahlt). "Anfängern" ist eine möglichst feingranulare Darstellung zu empfehlen, da ihre Schätzung dadurch zuverlässiger wird.

# Zeitplan: Meilensteine



22 / 58

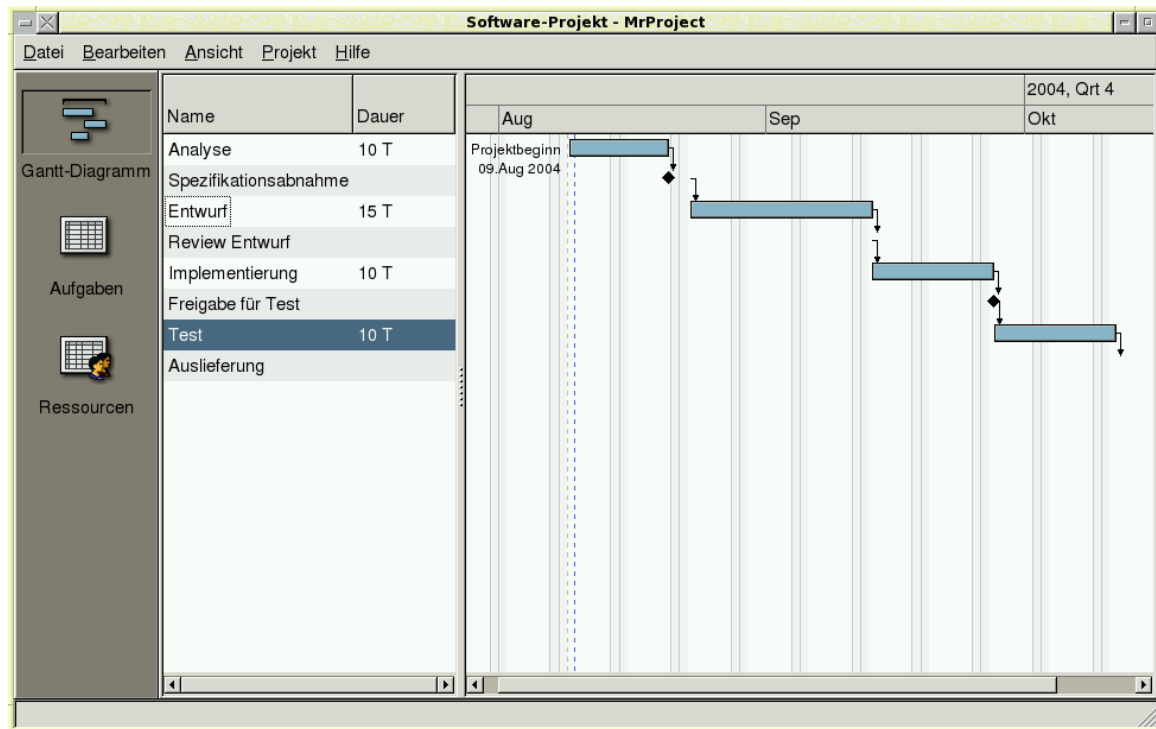
An verschiedenen Zeitpunkten des Projekts werden Meilensteine gesetzt.

Ein Meilenstein ist ein Zeitpunkt, zu dem ein prüfbares Ergebnis vorliegen muss.

Meilensteine erlauben die Beobachtung des Projektfortschritts. Die Projektverfolgung und die Qualitätssicherung werden interne Meilensteine setzen, also solche die nur dem Projekt bekannt sind, wie zum Beispiel das Review des Entwurfs. Zumindest bei jeder Übergabe eines Zwischenprodukts (Entwurfsdokument, Quellcode etc.) sollte ein Meilenstein definiert werden, der die Übergebenden und Empfänger des Zwischenprodukts einbezieht. Der Meilenstein legt die Kriterien für eine erfolgreiche Übergabe fest.

Externe Meilensteine involvieren den Kunden, z.B. die Abnahme der Spezifikation und der Akzeptanztest.

# Zeitplan: Abhängigkeiten

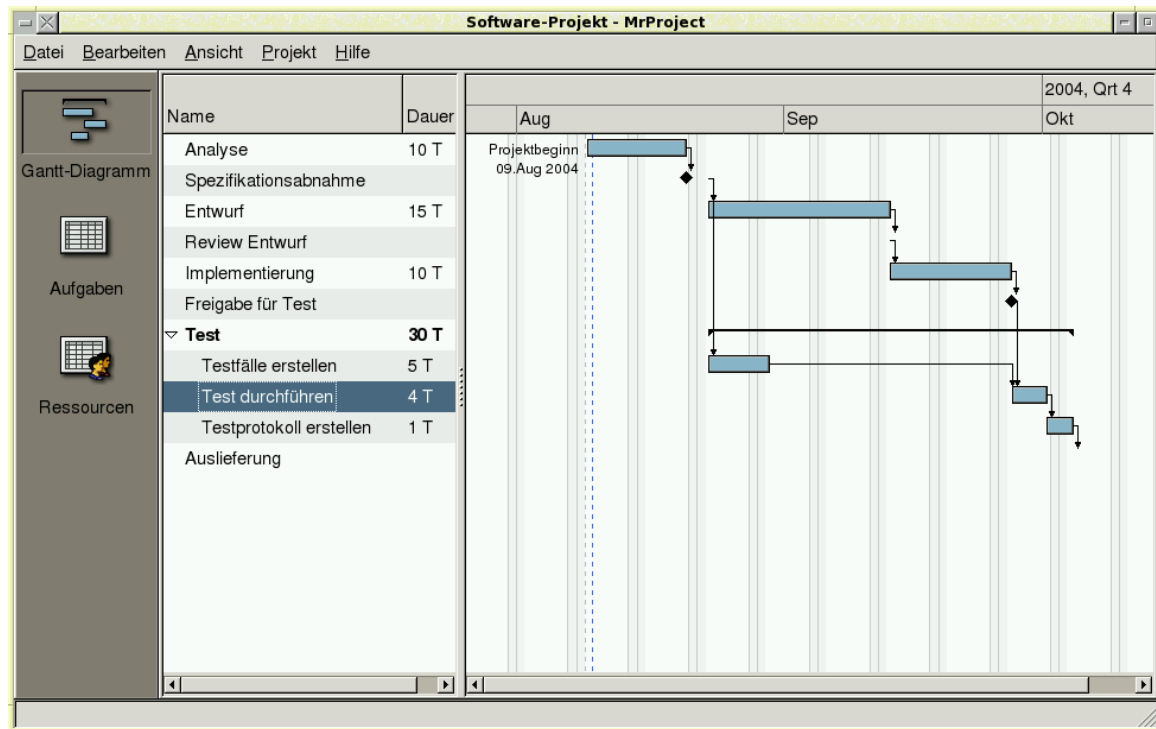


23 / 58

Zwischen den Aktivitäten existieren kausale Abhängigkeiten, die festgelegt werden müssen. Beispielsweise kann nur getestet werden, wenn der Quellcode existiert.

Die kausalen Abhängigkeiten führen zu einer partiellen Sequenzialisierung der Aktivitäten.

# Zeitplan: Verfeinerung

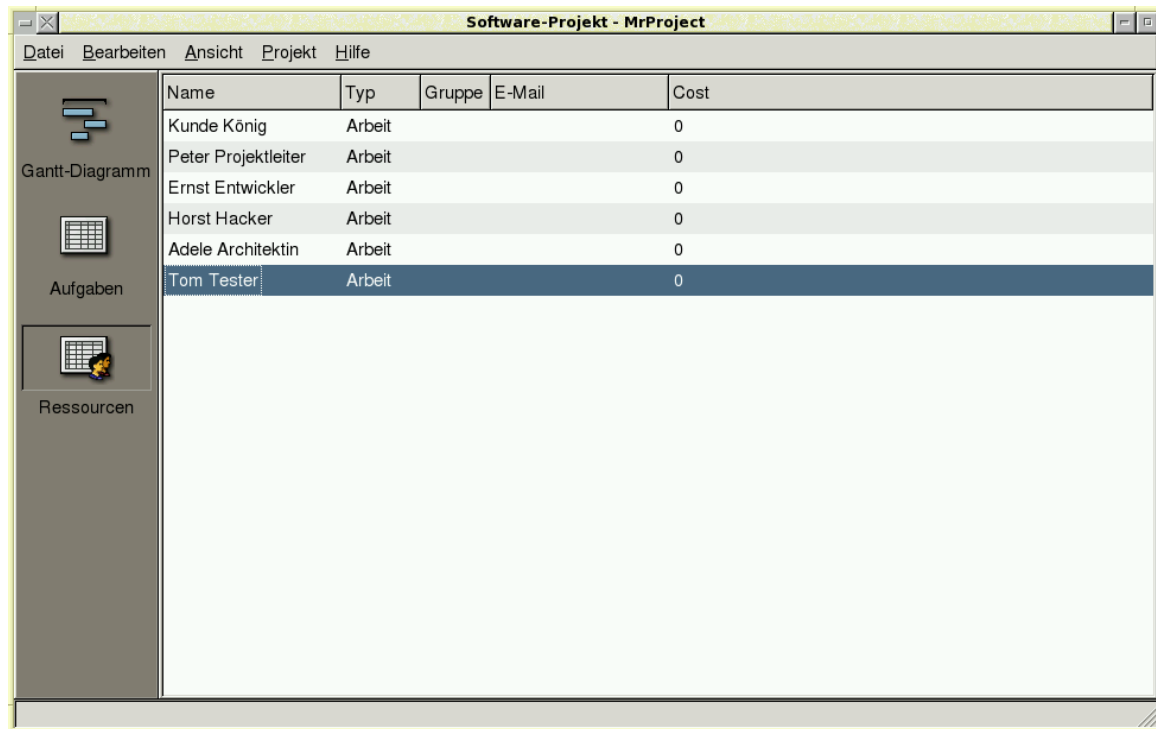


24 / 58

Grobgranulare Aktivitäten sollten verfeinert werden (siehe oben). Die Verfeinerung erlaubt auch eine getreue Darstellung der Abhängigkeiten zwischen den Aktivitäten. So können zum Beispiel Testfälle für den Black-Box-Test (siehe das spätere Kapitel über Tests) bereits beim Vorliegen der Spezifikation vorbereitet werden.

Dadurch kann die Möglichkeit zur Parallelisierung der Aktivitäten erhöht werden.

# Zeitplan: Ressourcen



Name	Typ	Gruppe	E-Mail	Cost
Kunde König	Arbeit			0
Peter Projektleiter	Arbeit			0
Ernst Entwickler	Arbeit			0
Horst Hacker	Arbeit			0
Adele Architektin	Arbeit			0
Tom Tester	Arbeit			0

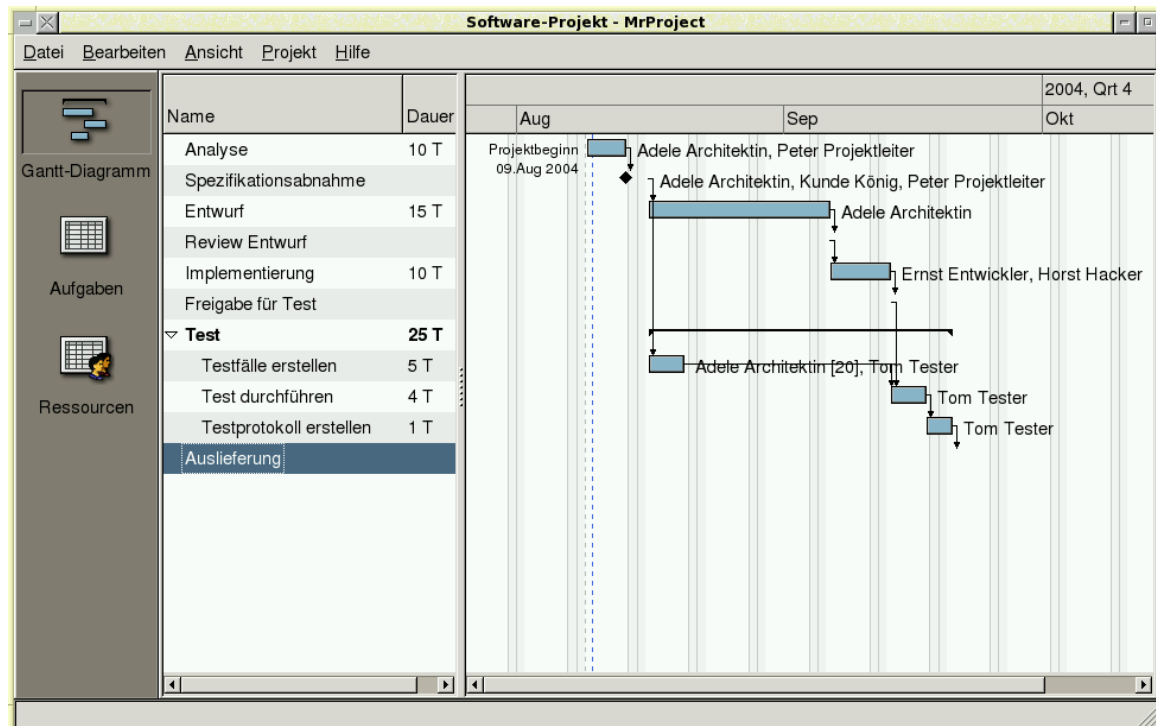
25 / 58

Ressourcen im Software-Projekt sind in der Regel menschliche Wesen. Bei der Entwicklung eingebetteter Systeme ist z.B. auch die Zielhardware eine Ressource, die eingeplant werden muss. In vielen Fällen wird sie parallel zur Software entwickelt.

Der Kunde ist eine wichtige Ressource, die eingeplant werden muss, und selten verfügbar ist.

Der Begriff "Ressource" auf Menschen angewandt klingt sehr technokratisch. Das ist hier nicht so gemeint.

# Zeitplan: Einplanung der Ressourcen



26 / 58

Schließlich werden die Ressourcen den Aktivitäten zugewiesen. Dadurch kann sich die Dauer von Aktivitäten verlängern oder verkürzen. Sie verlängern sich zum Beispiel, weil ein Entwickler auch für quasi-zeitgleiche Aktivitäten eingeplant wurde und selbstverständlich eine begrenzte wöchentliche Arbeitszeit hat. Sie verkürzt sich, wenn mehrere Entwickler einer Aktivität zugeordnet werden. Aber Achtung! Im Gegensatz zu anderen Projekten gilt im Software-Projekt nicht:

$$Dauer = \frac{Aufwand}{Anzahl\ Personen}$$

Je mehr Personen an einer Aktivität beteiligt sind desto höher ist der Aufwand für Kommunikation und Abstimmung. Statt dessen gilt der Rat: Weniger und bessere Leute nehmen!

Ein noch häufiger Irrglaube ist, dass man durch späte Hinzunahme zusätzlicher Leute einen in Schieflage geratenen Plan noch einhalten kann. Der Aufwand erhöht sich dann nicht nur durch erhöhte Kommunikation, sondern auch durch die Einarbeitung der Neuen.

# Kritischer Pfad

## Definition

**Kritischer Pfad:** die von der Dauer her längste Kette von Aktivitäten.

27 / 58

Bedeutung des kritischen Pfads:

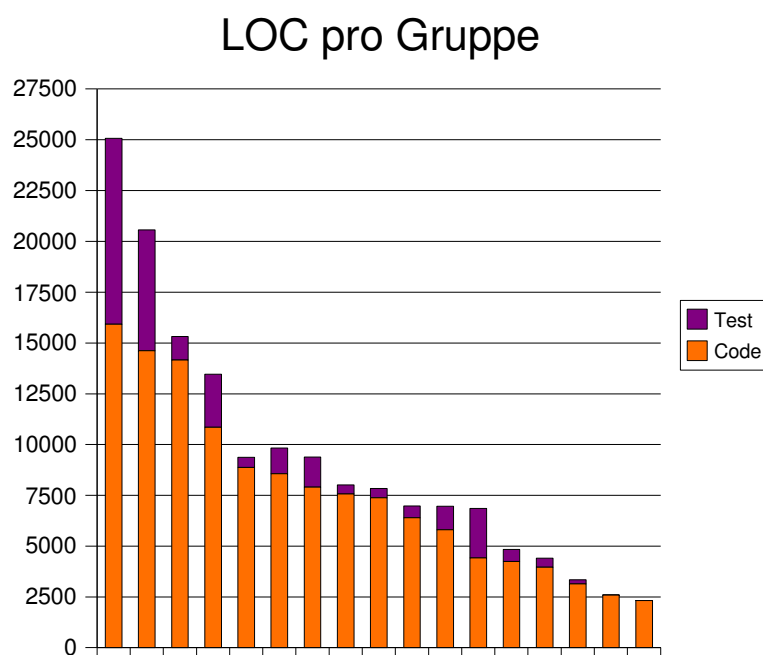
- bestimmt die Dauer des Projekts;
- Verspätungen in dieser Kette schlagen sich auf die Dauer des Projekts nieder;
- muss während des Projektverlaufs stets im Auge behalten werden.



Wie kann ich als Anfänger vernünftig schätzen?

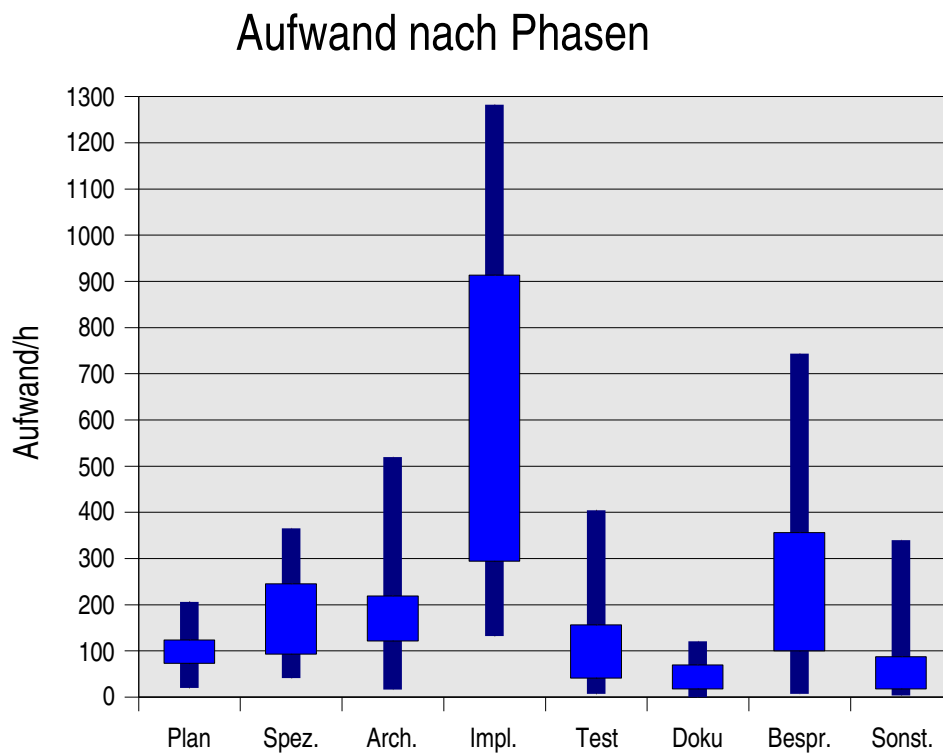
28 / 58

## Geschriebene Codezeilen im SWP 05/06



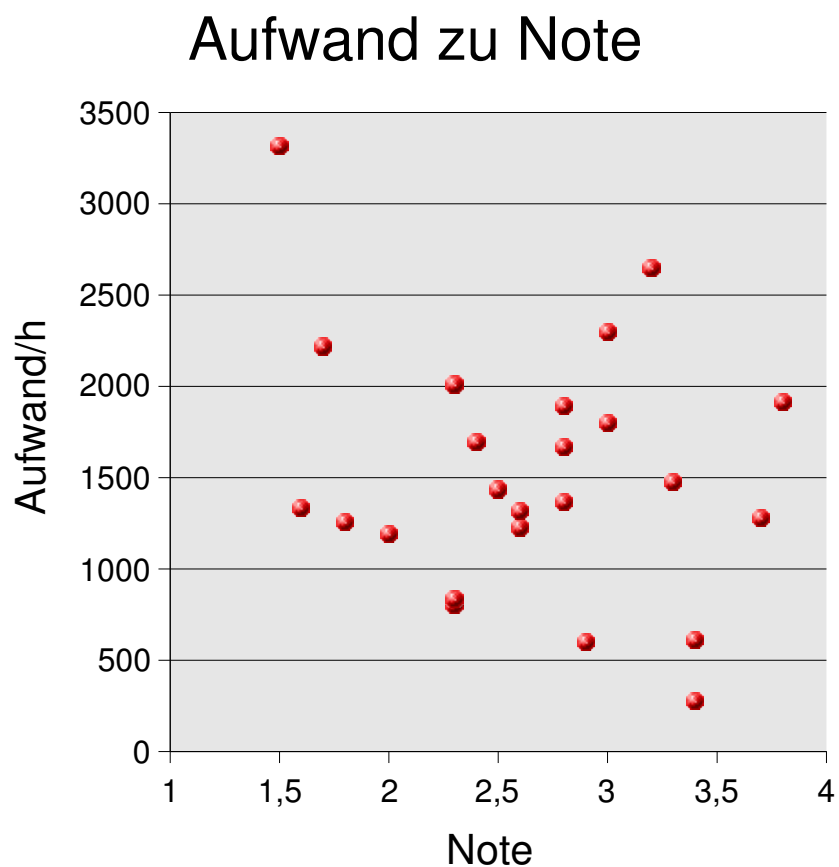
29 / 58





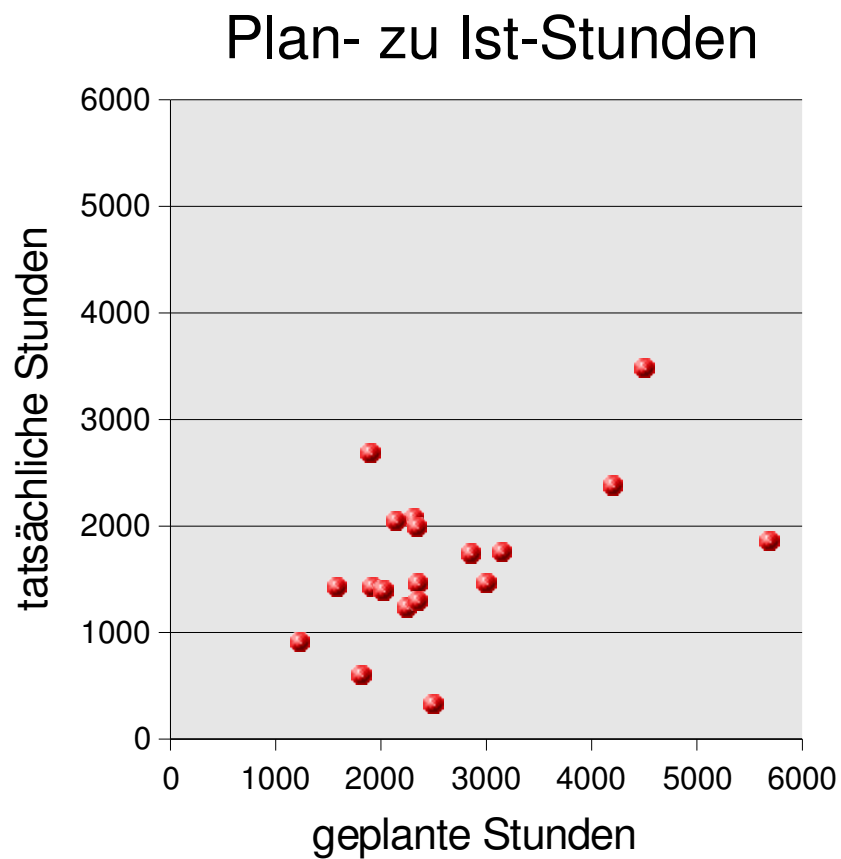
30 / 58

## Note versus Aufwand



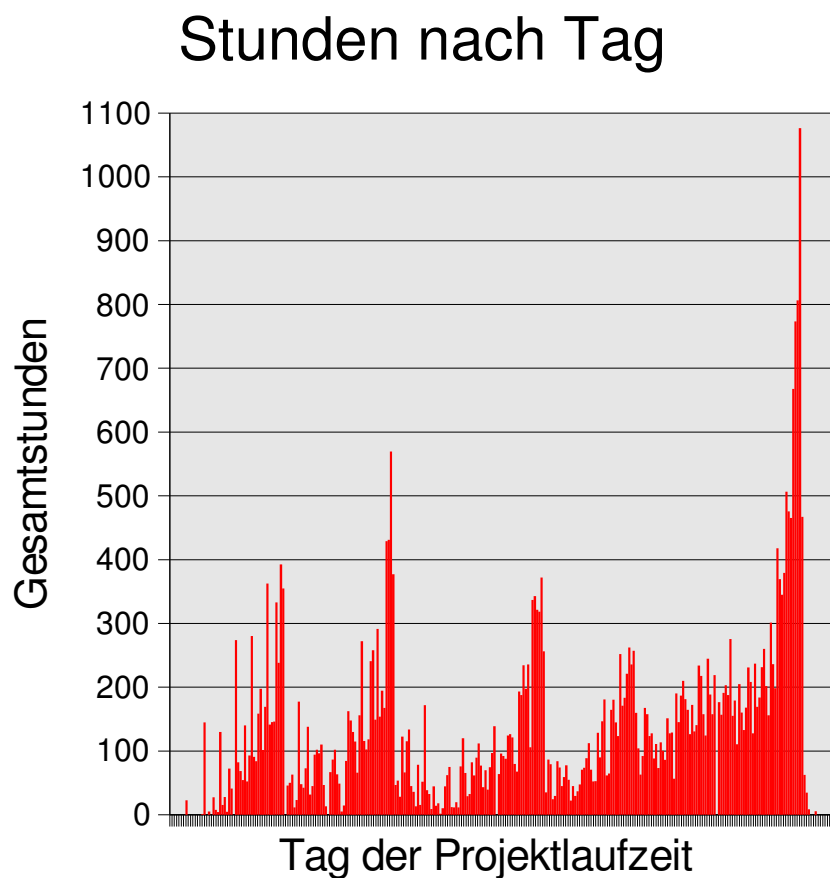
31 / 58

## Planung: Soll und Ist



32 / 58

## Arbeitsstunden pro Tag



33 / 58

# Fragen



- Warum soll ich im Projektplan Aussagen zu Risiken machen?
- Was ist denn Risikomanagement?
- Mit welchen Risiken muss ich rechnen?

35 / 58

# Risiken

*If you do not actively attack the risks in your project,  
they will actively attack you.*

– Gilb (1988)

## Definition

Ein **Risiko** ist ein Ereignis, dessen Eintreten den Erfolg des Projekts entscheidend behindern kann.

**Quantifizierte Risikohöhe** =

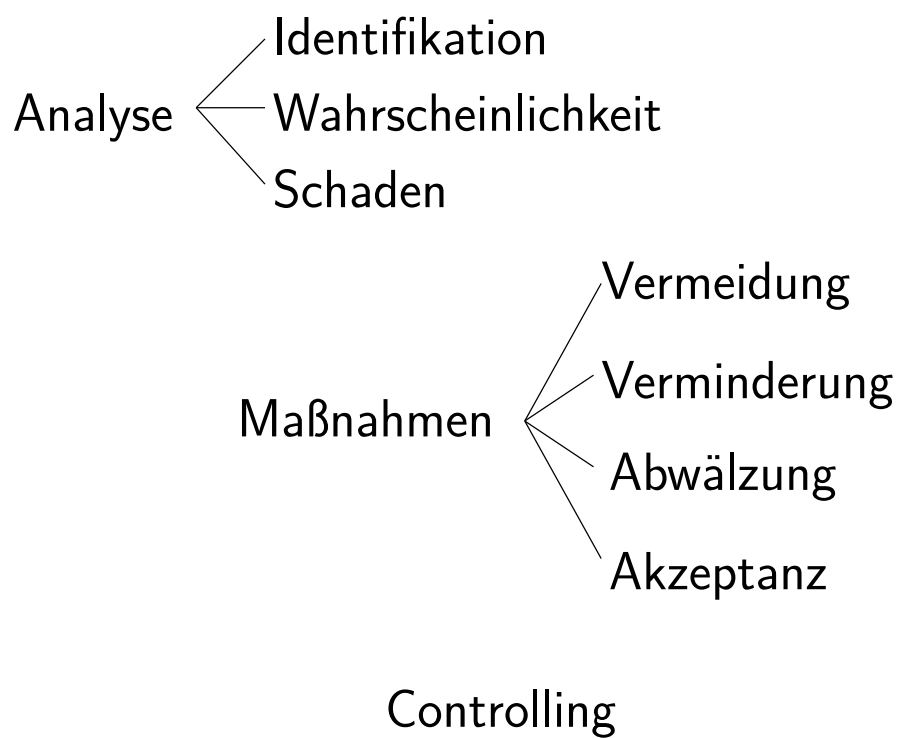
Eintrittswahrscheinlichkeit × maximale Schadenshöhe

36 / 58

Risikomanagement kümmert sich um Risiken, bevor sie auftreten.

Krisenmanagement kümmert sich um aufgetretene Risiken.

## Risiko-Management



## Risikoanalyse

- Identifizierung der Risiken
- Einschätzung der Eintrittswahrscheinlichkeit
- Einschätzung des potentiellen Schadens

## Risikomaßnahmen: Planung von Gegenmaßnahmen

- Vermeidung: Unterlassung der Aktivität
- Verminderung/Begrenzung auf akzeptables Maß, z.B. Streuung oder Haftung bis zu einem Limit
- Abwälzung: z.B. Unterauftragnehmer
- Akzeptanz, wenn Kosten/Nutzen im Verhältnis stehen

## Risiko-Controlling: Monitoring während des Projekts

## Risikoquellen

### Produktimmanent: Produkt und Technologie

- z.B. ungenaue Anforderungen
- z.B. instabile Anforderungen

### Projektimmanent: Mensch, Vorgehen, Management/Organisation und Infrastruktur

- z.B. Ressourcenprobleme
- z.B. mangelnde Motivation

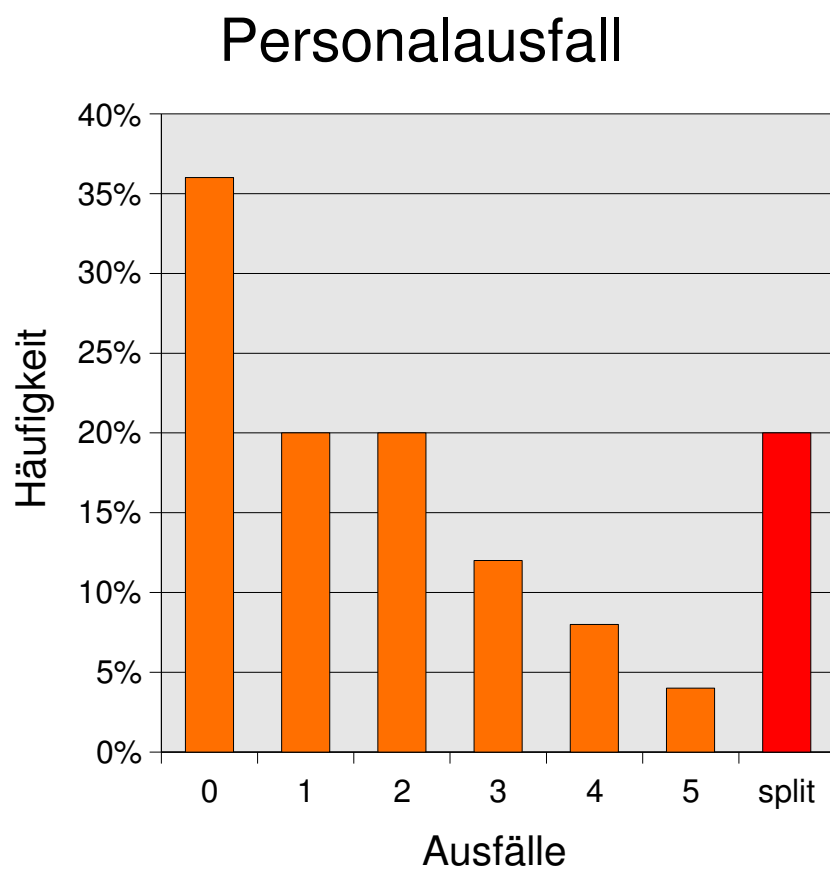
### Umfeldabhängig: Organisatorische Einbettung, Politik, Recht und Kultur

- z.B. Verschiebung von Prioritäten
- z.B. Abhängigkeit zu anderen Projekten

→ Historie: aus abgeschlossenen Projekten lernen

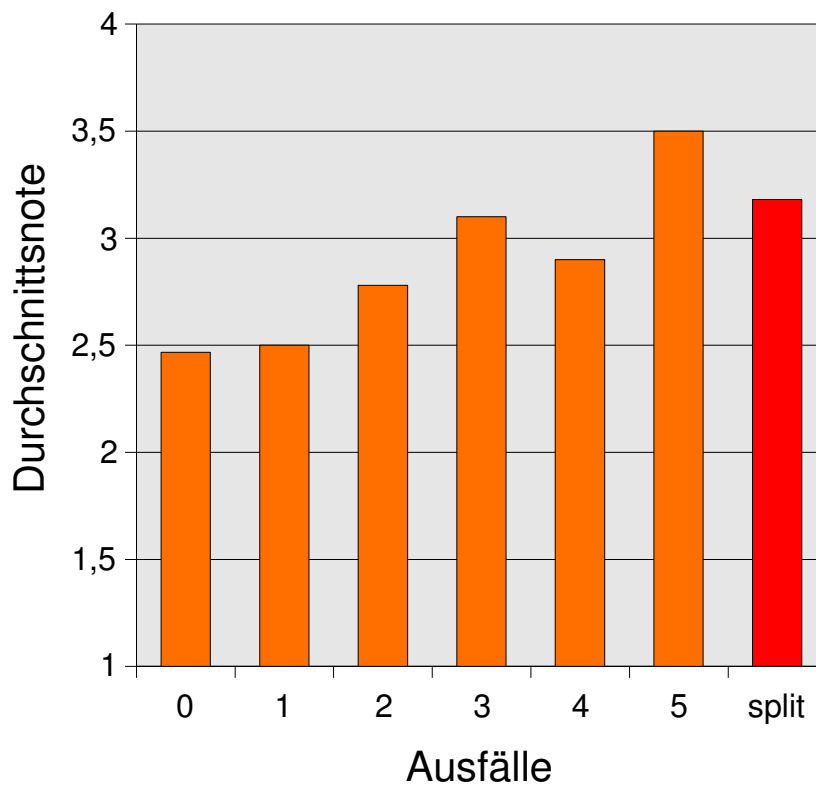
Zu Risiken und Nebenwirkungen fragen Sie Ihren  
... gesunden Menschenverstand, erfahrene Software-Entwickler, die Tageszeitung, die Literatur über  
Softwaretechnik ...

## Personalausfall



## Note versus Personalausfall

### Note nach Personalausfall



41 / 58

## Fragen



Der Projektabbruch schwebt als Risiko immer über uns. Ist der Abbruch in der Praxis aber immer fatal?

42 / 58

31% aller Software-Projekte werden vor Abschluss abgebrochen; weitere 53% sprengen den Zeit- oder Kostenrahmen oder liefern nicht die volle Funktionalität (Standish Group 1994)<sup>1</sup>

Gilt: Abbruch = Misserfolg wegen mangelhaftem Management?

---

<sup>1</sup>Dieser Bericht ist nicht unumstritten. Es gibt eine Reihe anderer Untersuchungen mit unterschiedlichen Ergebnissen Buschermöhle u. a. (2006); Sauer und Cuthbertson (2003); Standish Group (2004)

Die Statistik über den Abbruch von Projekten legt die Folgerung nahe, dass die Projekte abgebrochen wurden, weil sie schlecht geführt wurden. Diese Implikation ist falsch und gefährlich. Sie ist falsch, weil viele gut geführte Projekte abgebrochen werden, weil sich ihre ursprünglichen Annahmen geändert haben. Sie werden aus vertretbarem Grund beendet. Dies ist vor allem in Feldern mit schnellem Wandel häufig anzutreffen.

Die Implikation ist gefährlich, weil Projektmanager sich der Verführung konfrontiert sehen, ein eigentlich obsoletes Projekt weiterzuführen, um nicht als Versager dazustehen. Es gibt oft gute, vertretbare Gründe, ein Projekt abzuberechnen. Der Abbruch ist dann weiser als die Fortführung.



# Gründe für Projektabbruch nach Boehm (2000)

Unvollständige Anforderungen (13 %<sup>2</sup>):

Ursache bei	
<b>schlecht</b> geführtem Projekt	<b>gut</b> geführtem Projekt
meistens	häufig
Projekt startet ohne klare Idee der Bedürfnisse und Prioritäten der Stakeholder.	Stakeholder können sich nicht auf Anforderungen einigen.

---

<sup>2</sup>relativ zu den abgebrochenen Projekten

44 / 58

Die angegebenen Zahlen hat Barry Boehm anhand von Projekten ermittelt, bei denen er selbst involviert war (5-6 Projekte für digitale Bibliotheken pro Jahr, Begutachtung von ungefähr 20 Berichten über abgebrochene Industrieprojekte, unter Beteiligung von den 30 Angehörigen des Center for Software Engineering, das er leitet.

Andere Autoren berichten von Abbruchraten mit 40% und 50%, insbesondere für Gebiete, in denen neue Produkte eingeführt werden (Hayes (1997)).

Die meisten hier genannten Ursachen für Abbrüche stellen handfeste Risiken für **Ihr** Software-Projekt dar, denen ihr euch bewusst sein solltet. Die Fehler, die bei schlecht geführten Projekten gemacht werden, solltet ihr vermeiden.

Für euch folgt hier: Macht die Anforderungen und Prioritäten des Kunden fest, bevor ihr anfangt zu entwerfen und zu implementieren.

# Gründe für Projektabbruch nach Boehm (2000)

Mangelhafte Einbeziehung der Benutzer (12 %):

Ursache bei	
<b>schlecht</b> geführtem Projekt	<b>gut</b> geführtem Projekt
gleich häufig	
Projekt kommuniziert nicht mit Benutzer.	Benutzer kommuniziert nicht mit Projekt.

45 / 58

Für euch folgt hier: Sucht euren Kunden auf. Bezieht ihn bei der Anforderungsanalyse ein. Führt eine anständige und umfassende Ist-Analyse durch. Lasst den Kunden das Pflichtenheft prüfen (verwendet dabei eine ihm passende Terminologie; definiert ein Begriffslexikon). Entwickelt Prototypen, die dem Kunden vorgeführt werden. Haltet Kontakt mit dem Kunden auch während der Implementierungsphase. Prüft periodisch, ob sich seine Prioritäten und Anforderungen geändert haben.

# Gründe für Projektabbruch nach Boehm (2000)

Mangel an Ressourcen (11 %):

Ursache bei	
<b>schlecht</b> geführtem Projekt	<b>gut</b> geführtem Projekt
gleich häufig; schlecht geführte Projekte haben jedoch niedrigeren Geschäftswert und sind tendenziell eher betroffen	
Projekt hat wenig Geschäftswert. Budgeteinschnitte, Verkleinerungen, Repriorisierungen.	

46 / 58

Für euch folgt hier: Eure Ressource ist im Wesentlichen die Zeit; insbesondere auch die Zeit eurer Mitstreiter im Projekt. Deren Prioritäten sind nicht immer verträglich mit euren Prioritäten. Manche Projektteilnehmer werden es an Einsatz vermissen lassen.

Weitere potenziell mangelnde Ressourcen sind Rechner, Netzwerke und Software-Werkzeuge, die ihr für das Projekt benötigt.

# Gründe für Projektabbruch nach Boehm (2000)

## Unrealistische Erwartungen (10 %):

Ursache bei	
<b>schlecht</b> geführtem Projekt	<b>gut</b> geführtem Projekt
gleich häufig	
Machbarkeit wurde nicht geprüft.	Machbarkeitsprüfung fiel negativ aus.

47 / 58

Unrealistische Erwartungen des Benutzers liegen in dessen Natur. Er kennt selten technische Randbedingungen und Grenzen der Berechenbarkeit. Software gilt als beliebig flexibel: "Ich dachte, da müssen Sie nur ein Bit umkippen.". Und selbstverständlich will er stets noch einmal über den Preis reden.

Die Tutoren werden versuchen, den Benutzer realistisch zu simulieren.

Aber auch ihr selbst könnt zu unrealistischen Erwartungen neigen. Ihr seid möglicherweise nicht genug mit dem Anwendungsbereich vertraut oder unterschätzt den Aufwand an Kommunikation, den ein Mehrpersonenprojekt mit sich bringt, und den Aufwand für die Analyse, den Entwurf und den Test.

Letztlich steht euch circa ein Tag pro Woche für das Projekt zur Verfügung und diese Projektstage sind unterbrochen von vielen anderen Aktivitäten. Für euch folgt hier: Klopft frühzeitig die Anforderungen des Kunden auf Machbarkeit ab. Erstellt Prototypen, um die Machbarkeit kritischer Anforderungen zu überprüfen. Plant realistisch. Gebt nicht in allen Punkten dem Kunden nach, wenn dieser seine Wünsche äußert. Es gilt "quid pro quo": etwas für etwas. Will er X, muss er sich bei Y beschränken. Verlangt ihm Prioritäten ab.

# Gründe für Projektabbruch nach Boehm (2000)

Mangelnde Unterstützung bei der Ausführung (9 %):

Ursache bei	
<b>schlecht</b> geführtem Projekt	<b>gut</b> geführtem Projekt
meistens	häufig
Manager machen unverifizierte Annahmen über Unterstützung (z.B. verlassen sich darauf, dass andere Initiativen repriorisiert werden, um Projekt zu unterstützen).	Unterstützung wird entzogen (z.B. Verantwortliche werden ausgetauscht; neue Verantwortliche haben andere Prioritäten und Agenda).

48 / 58

Ihr müsst euch mit Mitgliedern eurer Gruppe und auch mit anderen Gruppen auseinandersetzen. Der Betreuer/die Betreuerin betreut mehrere Gruppen und hat noch viele andere Pflichten. Teilnehmer wie Betreuer sind möglicherweise zeitweise oder auch dauerhaft nicht verfügbar.

Für euch folgt hier: Seid explizit darüber, was ihr von anderen erwartet und auch bis wann ihr etwas erwartet. Kommuniziert!

# Gründe für Projektabbruch nach Boehm (2000)

Anforderungen ändern sich (9%):

Ursache bei	
<b>schlecht</b> geführtem Projekt	<b>gut</b> geführtem Projekt
meistens	nicht selten
Änderungen werden akzeptiert, ohne dass Budget und Projektplan angepasst werden.	Folgekosten der Änderung überwiegen den Nutzen des Projekts.

49 / 58

Die Anforderungen sind am Anfang nicht klar verstanden. Der Kunde selbst wird sich erst im Laufe des Projekt klarer darüber, was er eigentlich will. Rahmenbedingungen ändern sich.

Die Anforderungen sind selten als stabil zu betrachten.

Für euch folgt hier: Haltet vertraglich fest, wie ihr und der Kunde mit Änderungen umgehen wollt. Antizipiert mögliche Änderungen. Überlegt euch gut, welche Änderung ihr akzeptiert. Seid euch über die Konsequenzen einer Änderung im Klaren.

# Gründe für Projektabbruch nach Boehm (2000)

## Mangelhafte Planung (8%):

Ursache bei	
<b>schlecht</b> geführtem Projekt	<b>gut</b> geführtem Projekt
immer	—
Projektmanager haben keine Ahnung, wo sie sich befinden und wann das Projekt fertig wird.	

50 / 58

Ihr seid weder mit dem Anwendungsbereich noch mit einem Projekt dieser Art vertraut. Ihr werdet es besonders schwer haben.

Für euch folgt: Seid nicht zu optimistisch. Unterschätzt nicht das Problem, insbesondere in seinen Details, und überschätzt euch nicht selbst.

Plant und seid euch zu jedem Zeitpunkt darüber im Klaren, wo ihr euch tatsächlich befindet und was das eigentliche Planziel war. Seid vorsichtig mit dem Glauben: "Das holen wir später wieder ein".

Reagiert, wenn Soll und Ist zu weit auseinander klaffen. Sprecht frühzeitig mit eurem Kunden über mögliche Einschränkungen bei der Leistung. Der Kunde ist sehr wahrscheinlich besser bedient, wenn er wenigstens etwas bekommt, und nicht gar nichts. Das Vertrauensverhältnis wäre auf immer zerrüttet, wenn ihr ihn erst am Tag der Auslieferung über den wahren Zustand eures Projekts aufklärt.

Denkt an die Manager von TollCollect, die noch einen Monat vor der geplanten Einführung behauptet haben, sie würden den Termin halten.

## Gründe für Projektabbruch nach Boehm (2000)

Kein Nutzen (8%):

Ursache bei	
<b>schlecht</b> geführtem Projekt	<b>gut</b> geführtem Projekt
gleich häufig in Feldern mit schnellem Wandel	
	Gute Projektmanager verfolgen Trends und erkennen nachlassenden Nutzen früher; sie reagieren frühzeitiger mit Abbruch.

51 / 58

## Gründe für Projektabbruch nach Boehm (2000)

Mangelndes IT-Management (6%):

Ursache bei	
<b>schlecht</b> geführtem Projekt	<b>gut</b> geführtem Projekt
immer	—
Offensichtlich mangelhaftes Management.	

52 / 58



Geht in die Vorlesung. Die Vorlesung möchte euch genau hiervor bewahren. Kommt ihr nicht allein weiter, holt euch den Rat eines externen Consultants, sprich eures Tutors, ein.

Solltet ihr nicht selbst der Projektmanager sein, dann sprecht mit ihm offen über das Problem. Ein Projektleiter ist kein General, sondern ein Dienstleister für die Projektmitglieder. Er soll die Übersicht wahren und das Projekt zusammenhalten. Schafft er dies nicht, obwohl ihr mit ihm darüber geredet habt, dann wechselt ihn aus.

## Gründe für Projektabbruch nach Boehm (2000)

Mangelndes Verständnis der Technologie (4%):

Ursache bei	
<b>schlecht</b> geführtem Projekt	<b>gut</b> geführtem Projekt
allermeistens	—
Fehlende Kenntnisse der Entwickler und Manager; Projekte, die niemals hätten begonnen werden sollen.	

Bildet Experten, die sich mit spezifischen technischen Problemen gezielt auseinander setzen. Baut Prototypen in einer Technologie, die ihr noch nicht beherrscht. Dreht nicht gleichzeitig an zu vielen Schrauben: Nicht allen Technologien gleichzeitig hinterherlaufen, sondern eine nach der anderen inkrementell einführen. Betrachtet das Problem unbekannter Technologien frühzeitig in eurem Projektplan (räumt zusätzliche Zeit hierfür ein). Bedenkt, dass es nicht ausreicht, mehrere Technologien isoliert zu beherrschen. Ihr könntet euer blaues Wunder erleben, wenn ihr schließlich versucht, diese Technologien miteinander zu integrieren.

## Fragen



Die menschliche Seite: Was gilt es bei der Einschätzung von Risiken zu beachten?

# Menschliche Wahrnehmung von Risiken

- Selbstgewählte Gefahren erscheinen geringer als aufgezwungene.
- Prinzipiell kontrollierbare Risiken sind akzeptabler als solche, auf die wir scheinbar keinen Einfluss haben.
- Natürliche Risiken werden eher hingenommen als von Menschen geschaffene.
- Katastrophen alarmieren uns mehr als der alltägliche Wahnsinn.
- Risiken, die von schwer fassbaren Techniken ausgehen, werden eher wahrgenommen als die von vertrauten Techniken.
- Schlechte Nachrichten werden eher geglaubt als positive.

55 / 58

## Selbstgewählte Gefahren

- Allgemein: Die Risiken bestimmter Sportarten wie Skifahren oder Reiten gehen wir bewusst ein. Dagegen wehren wir uns gegen Konservierungsstoffe in der Nahrung.
- Softwaretechnik: Eine vom Kunden gewollte fremde Technologie erscheint uns riskanter als eine selbst gewollte fremde Technologie.

## Kontrollierbare Risiken

- Allgemein: Fett, nährstoffarmes Essen ist beliebt, während Leitungswasser auch dann gemieden wird, wenn die Trinkqualität garantiert ist.
- Softwaretechnik: Wir scheuen uns eine Software-Bibliothek zu verwenden und implementieren unsere Hash-Tabelle lieber selbst.

## Natürliche Risiken

- Allgemein: In der Erde vorkommendes Radon erscheint uns harmlos im Vergleich zur selben Strahlungsintensität aus künstlichen Quellen.
- Softwaretechnik: Rutschige Straßenverhältnisse erscheinen uns harmlos im Vergleich zu Fehlern in der Software des Bremssystems im Auto.

## Katastrophen alarmieren

- Allgemein: Werden nach einem Schiffsunglück Giftbeutel oder Ölkumpen an die Strände geschwemmt, ist die Aufregung groß, während die schleichende Vergiftung der Meere kaum zur Kenntnis genommen wird.
- Softwaretechnik: Der Projektabbruch alarmiert uns mehr als die schleichende Verschlechterung der Qualität, die wir ausliefern.

## Schwer fassbare Techniken

- Allgemein: Eine Müllverbrennungsanlage mit relativ geringen Emissionen wird bekämpft, der Autoverkehr hingegen verteidigt.
- Softwaretechnik: Die Einführung der Cleanroom-Development-Methode erscheint uns riskanter als die Code-and-Fix-Methode.

## Schlechte Nachrichten

- Allgemein: Stürme und Überschwemmungen gelten als Beweis für den Treibhauseffekt, doch die geringer gewordene Verschmutzung des Rheins halten viele für Propaganda der Industrie.
- Softwaretechnik: Das Gerücht, dass das Projekt in Schieflage geraten ist, macht hellhörig; eine Aussage, die Deadline werde eingehalten, glauben wir gerne.

## Wiederholungsfragen I

- Erstellen Sie einen Projektplan für ein Software-Projekt.
- Was sind die Elemente eines Projekts? Insbesondere was ist ein Meilenstein und eine Baseline?
- Wann wird geplant?
- Wie geht man beim Planen vor?
- Was ist der Inhalt eines Projektplans?
- Was ist ein Gantt-Diagramm?
- Was ist ein kritischer Pfad? Welche Bedeutung hat er?
- Was sind die Ressourcen eines Software-Projekts? Was sind deren Besonderheiten?
- Was sind typische Risiken in einem Software-Projekt? Wie geht man mit ihnen um?
- Was sind die Besonderheiten eines Software-Projekts?
- Welche Risiken sind typisch für Software-Projekte? Wie ist mit ihnen umzugehen?
- Erläutern Sie Risiko-Management.

56 / 58

- 1 Boehm 2000** BOEHM, Barry: Project Termination Doesn't Equal Project Failure. In: IEEE Computer (2000), September, S. 94–96
- 2 Buschermöhle u. a. 2006** BUSCHERMÖHLE, Ralf ; EEKHOFF, Heike ; JOSKO, Bernhard: Success – Erfolgs- und Misserfolgsk Faktoren bei der Durchführung von Hard- und Softwareentwicklungsprojekten in Deutschland.  
[www.offis.de/umfragesuccess](http://www.offis.de/umfragesuccess). 2006
- 3 Gilb 1988** GILB, Tom: Principles of Software Engineering Management. Harlow, UK : Addison-Wesley, 1988
- 4 Hayes 1997** HAYES, Frank: Managing User Expectations. In: Computerworld (1997), November
- 5 IEEE-Std-1058 1987** : ANSI/IEEE Standard for Software Project Management Plans. ANSI/IEEE Std. 1058.1-1987. 1987

57 / 58

- 6 Sauer und Cuthbertson 2003** SAUER, C. ; CUTHBERTSON, C.: The State of IT Project Management in the UK. <http://www.cw360ms.com/pmsurveyresults/surveyresults>. 2003
- 7 Standish Group 1994**
- 8 Standish Group 2004** STANDISH GROUP: Third Quarter Research Report. <http://www.standishgroup.com>. 2004